Pre-class activities
Support Vector Machines and Kernels

Emmanuel Rachelson

1 Quadratic programming

Suppose a collection of N pairs $\{(x_i, y_i)\}_{i=1..N}$ with $x_i \in \mathbb{R}^n$ and $y_i \in \{-1;1\}$ and consider the following optimization problem:

$$\min_{w \in \mathbb{R}^n} \frac{1}{2}\|w\|^2$$
$$s.t.\ \forall j = 1...N, \quad y_i \left(w^T x_i\right) \leq 1$$

☞ Recall the name of this type of optimization problem.

☞ Write down the problem's Lagrangian.

☞ Why are the constraints qualified?

☞ Write Karush-Kuhn-Tucker's first order conditions. Recall what they mean.

☞ Recall the duality theorem in Differentiable Optimization and write the dual form of the above optimization problem.

2 Insensitive Least squares regression

Suppose a collection of N points $\{(x_i, y_i)\}_{i=1..N}$ with $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$ and consider the associated regression problem. Suppose also that a family of functions $\Phi = \{\phi_j\}_{j=1..p},\ \phi_j : \mathbb{R}^n \to \mathbb{R}$ is provided and that the solution to the regression problem should lie in the space spanned by Φ (that is, the regression function f should have the form $f = \sum_{j=1}^{p} w_j \phi_j$).

☞ Write the regression problem as a least squares minimization problem.

Consider the data plotted in Figure 1.

Apart from the outlier at $x = 1.05$, the data fits the $y = x$ relation perfectly. In this case, advanced feature functions do not seem necessary, so the jth feature is actually the jth component of x. Thus the problem is a simple $y = w^T x$ regression problem.

☞ Can you think of a formulation of the regression problem that would be robust to noise? For example, that would discard any noisy point that stays inside a "tube" of width ϵ around the inferred function?

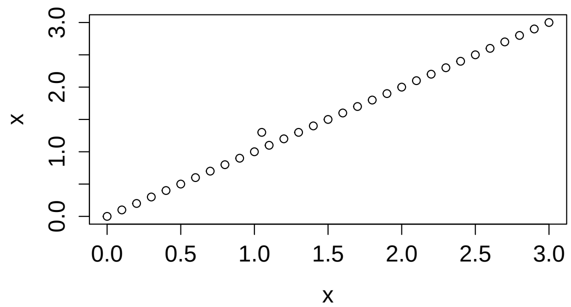

Figure 1: Regression data

3 The trick of the additional dimension

Consider the following test data where x is a voltage measurement and y indicates whether an electronic component failed under that voltage:

x	0.3	0.7	1.1	1.8	2.5	3.0	3.3	3.5	3.7
y	-1	-1	-1	1	1	1	1	-1	-1

☞ Figure 3 shows a graphical display of the above data set. One wishes to linearly separate the data. What is the (very naive) general form of a linear classifier on this data ? What is the best training error one can obtain with such a classifier ?

A smart engineer decides to plot the same data but enriches the description by adding a second axis representing $(2-x)^2$. The data set becomes:

$z_1 = x$	0.3	0.7	1.1	1.8	2.5	3.0	3.3	3.5	3.7
$z_2 = (2-x)^2$	2.89	1.69	0.81	0.04	0.25	1	1.69	2.25	2.89
y	-1	-1	-1	1	1	1	1	-1	-1

The new graphical representation is displayed on Figure 3.

☞ Did that operation seem to help the linear classification task? What lessons can one draw?

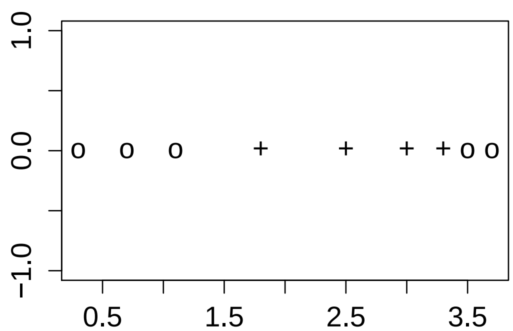

Figure 2: Raw measurements

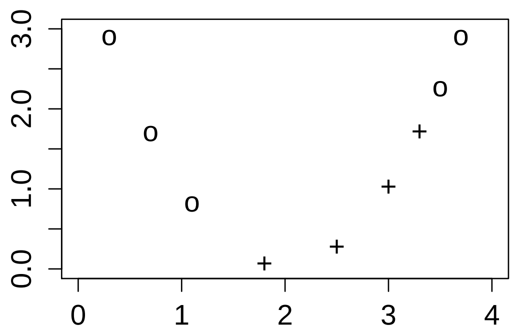

Figure 3: Enriched representation